

Habin LEE, *et al.*
Serial No. 10/594,424
March 6, 2009

AMENDMENTS TO THE DRAWINGS:

Applicants submit concurrently herewith eight (8) sheets of annotated drawings illustrating Figs 1(a), 1(b), 1(c), 1(d) and 2-8 with changes shown in red ink, accompanied by eight (8) sheets of replacement drawings incorporating the amendments.

Attachments: Replacement Sheets: 8
Annotated Sheets Showing Changes: 8

Habin LEE, *et al.*
Serial No. 10/594,424
March 6, 2009

REMARKS/ARGUMENTS

Reconsideration of this application is respectfully requested.

The correction to the specification requested by the Examiner has been effected by the above amendment – together with other amendments putting this application in more traditional U.S. format.

The rejection of claims 1-9 under 35 U.S.C. §101 because they are allegedly directed to non-statutory subject matter is respectfully traversed.

Claim 1 has been amended so as to “tie” it to another statutory class, i.e., a “machine.” In particular, claim 1 has been amended so as to require its steps to be implemented by using at least one computer-programmed processor, the multi-agent system having been provided within the operating environment of at least one computer-programmed processor having a CPU communicating with memory and input/output ports – as would be understood by one skilled in the art reading this original application at the time it was filed. The specification has been similarly amended at page 13 and this is not believed to constitute any “new” matter. See, for example, original claims 10-25 including, especially, claim 25 which constitute part of the originally filed “specification” under U.S. law.

Accordingly, all outstanding formality issues are now believed to have been resolved in the applicants’ favor.

The rejection of call claims 1-25 under 35 U.S.C. §102 as allegedly anticipated by Ahn *et al.*, 2003 (“a flexible agent system for change adaptation in supply chains” – hereinafter designated by “FAS”) is respectfully traversed.

Claims 1 and 10 have been amended (and dependent claims 5 and 16 correspondingly cancelled) to further clarify that the present invention provides a method and a software agent for processing and ultimately executing a service request in terms of a conversation model received by a software agent, when the received conversation model is not previously known by the software agent.

When the software agent receives a service request in terms of a known conversation model, the conversation model is implemented in conventional fashion. See page 16, lines 18-22 of the specification:

“...the conversation scheduler 515 at STEP 625 arranges for the model manager 505 to check whether the conversation policy referenced by the matching SD is known to the agent 500. If, at step 625, the matching SD references a known conversation policy, then at STEP 640 the conversation scheduler 515 simply arranges to execute the CM and at STEP 645 returns the service result arising from execution of that CM.”

As will be explained in more detail below, FAS merely discloses this conventional aspect.

Conventionally, including FAS, there is no mechanism for a software agent to properly process a service request in terms of an unknown received conversation model, i.e. one that is not already known to, and stored by, the receiving software agent at its respectively associated processor – but nevertheless is processed using its own resources! However, the present invention provides a novel and non-obvious way for the software agent to process a service request in terms of an unknown received conversational model. See page 16, lines 23-35 of the specification:

“Otherwise, if, at STEP 625 the CM is not known to the agent 500, then at STEP 630 the conversation scheduler 515 contacts the mediator agent again, via the model manager 505, to obtain a CM for the referenced conversation policy and at STEP 635 triggers the model manager 505 to determine whether or not the obtained CM is executable by the agent 500. If, at STEP 635, it is executable, then at STEP 640 the conversation scheduler 515 arranges to execute it, and at STEP 645 returns the service result arising from its execution. Otherwise, at STEP 650, the conversation scheduler 515 tries to access another service agent’s SD and processing returns to STEP 615.

The check to be performed by the model manager 505, at STEP 635, as to whether the agent 500 can execute the CM is based upon a determination of whether the ontology items used in the CM can be provided or processed by the agent 500. A preferred process by which the model manager 505 may make this determination at STEP 635 will now be described with reference to the flow chart shown in Figure 7.”

Thereafter, one embodiment describing one way in which the present invention processes such a received service request in terms of an unknown conversation model is then described with reference to Fig. 7, and in corresponding fashion the underlying invention (of which Fig. 7 is a specific embodiment) is claimed in claims 1 and 10. In particular, referring now to claim 1, after having received the unknown conversation model at step (i), the software agent:

at step (ii) identifies ontology items used in the conversation model;
at step (iii) determines, for each identified ontology item, whether the software agent can nevertheless provide or otherwise process the identified ontology item; and, if it can, then
at step (iv) it executes the conversation model.

In other words, the software agent goes ahead and executes the received conversation model, despite not previously knowing it as such and never having being sent specification information about it. This is done by virtue of a completely novel and non-obvious approach comprising individually identifying the various different ontology items used in the conversation model (at STEP 705 of Fig. 7, see the specification page 17, lines 2-4), then satisfying itself that it already has all these ontology items stored (at STEPS 720 and 725, see the specification page 17, lines 11-15) or alternatively can at least obtain them by a recursive process from stored behaviors (at STEPS 730 and 735, see the specification at page 17, lines 15-18, and Fig. 8 and associated text, Fig. 8 expounding STEP 730).

This process based on looking at ontology items of an unknown conversation model is novel and non-obvious over FAS. For example, FAS does not carry out any of the above-mentioned steps (i) to (iv), nor does FAS relate to, or teach, any means for a software agent to process a received service request in terms of a conversation model that is not already known to the software agent.

In contrast, FAS teaches a flexible agent system for adapting to changes in supply chains and the like. FAS teaches particular formats of conversation model that are convenient for easily providing new versions of conversational model.

For example, as discussed in a portion of FAS relied upon by the Examiner, i.e., the third paragraph of section 3.3 and associated Fig. 6, two new conversation models are produced, namely, “OrderRequestNewCPU” and “OrderRequestBarebone”. FAS teaches that these two new conversation models are produced as flexible adaptations of the simpler conversation model “SimpleOrderRequest” (also shown in Fig. 6). FAS teaches (see, e.g., page 607, left-hand column) that these two new conversation models are produced in response to business changes, i.e., “Case 1” a new supplier’s requirements, and “Case 2” a new type of product.

Significantly, the new conversation models are sent to relevant software agents, i.e., the new conversation models are exchanged and acquired between software agents (see section 3.2, where FAS teaches that, “*The key idea of FCM is to enable agent systems to exchange and acquire new conversation policies*”).

In other words, FAS (including section 3.4 relied upon by the Examiner with respect to original claims 5 and 16 directed to unknown conversation model handling) merely teaches acquiring some new conversation models, but these are nevertheless fully sent to and stored (acquired) at the software agents. Therefore, when a software agent receives a service request in terms of one of these new conversation models, that new conversation model becomes already known by, and stored at, the software agent, and hence the software agent can process the service request in conventional fashion. As outlined earlier above, this is not the same as the invention as claimed in amended claims 1 and 10.

Given such fundamental deficiencies of FAS with respect to independent claims 1 and 10, it is not necessary at this time to discuss additional deficiencies of FAS with respect to other aspects of the rejected claims. Suffice it to note that, as a matter of law, it

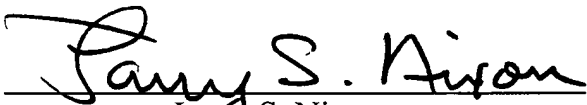
Habin LEE, *et al.*
Serial No. 10/594,424
March 6, 2009

is impossible for any reference to anticipate a claim unless it teaches each and every aspect of that claim.

Accordingly, this entire application is now believed to be in condition for allowance, and a formal notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
Larry S. Nixon
Reg. No. 25,640

LSN:lef

901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

1/8

ANNOTATED MARKED UP DRAWINGS
FOR SN 10/594,424

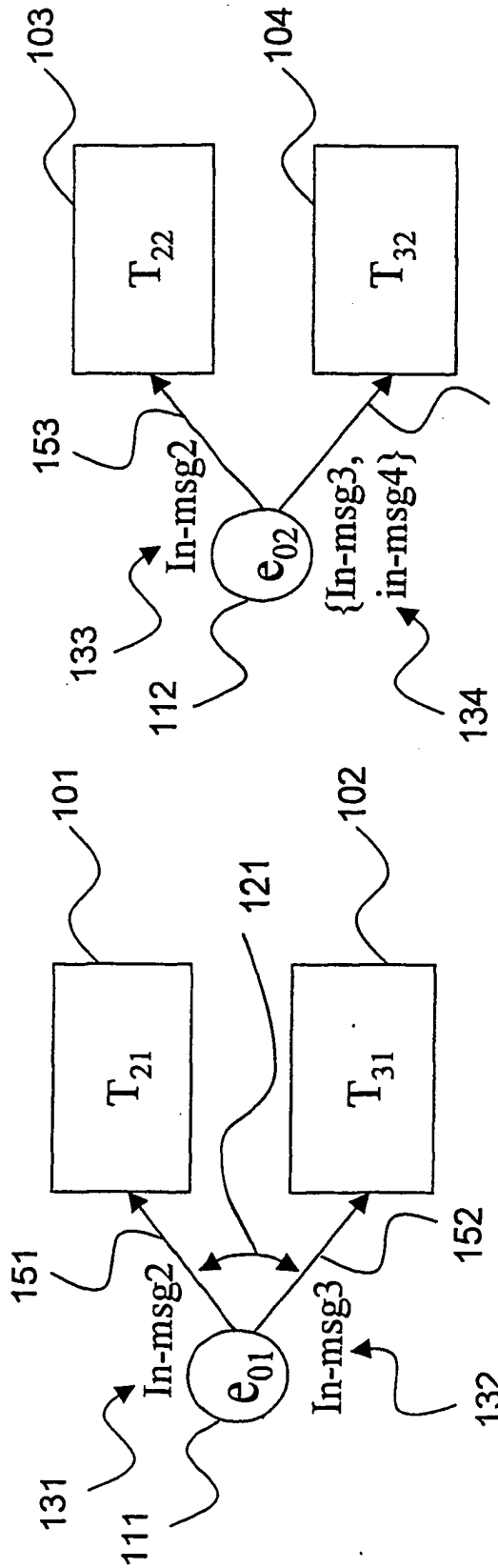


Fig 1(a)

Fig 1(b)

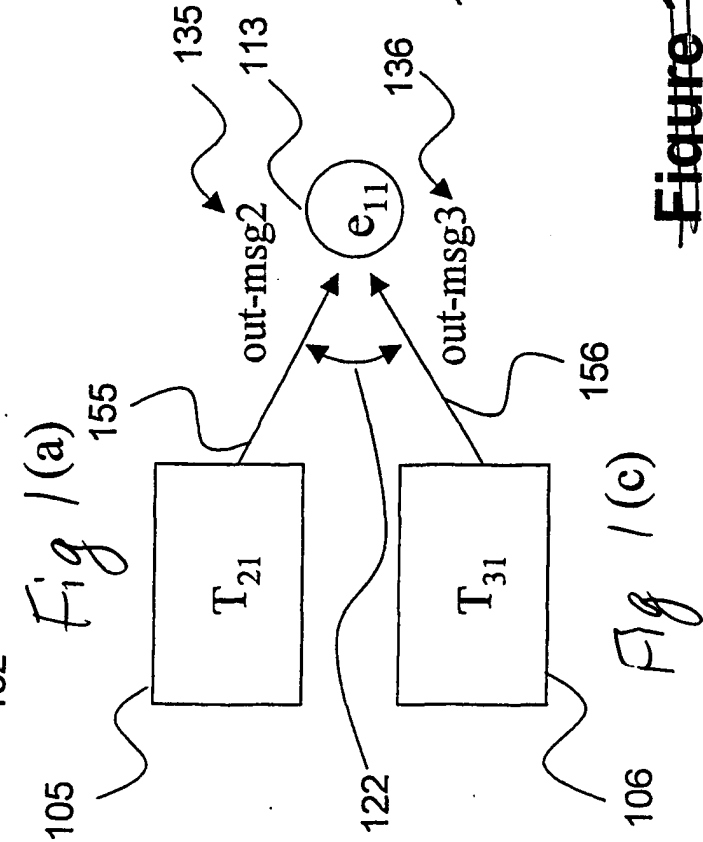


Fig 1(c)

Fig 1(d)

~~Figure 1~~

ANNOTATED MARKED UP DRAWINGS
FOR SN 10/594,424

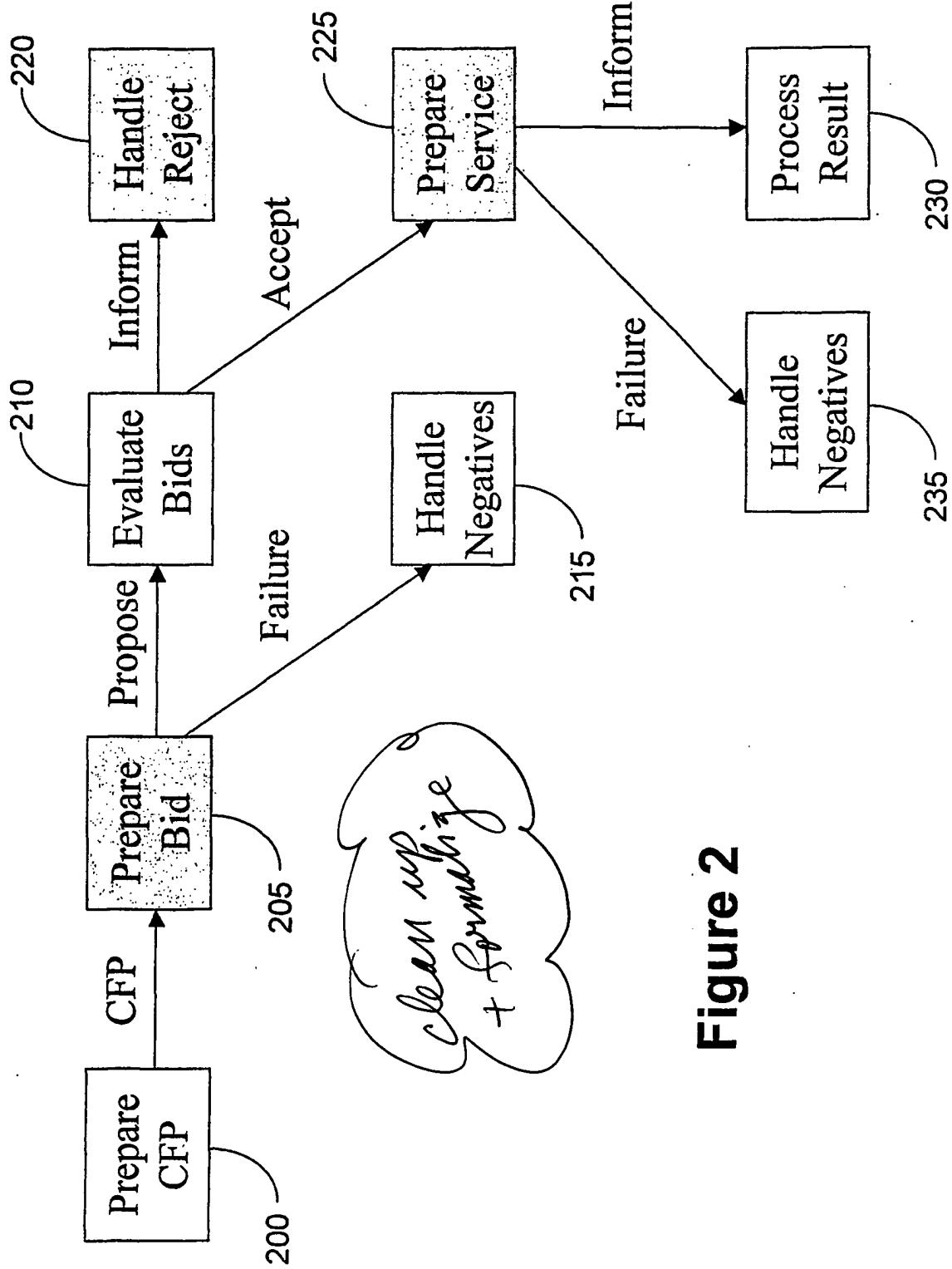


Figure 2

ANNOTATED MARKED UP DRAWINGS
FOR SN 10/594,424

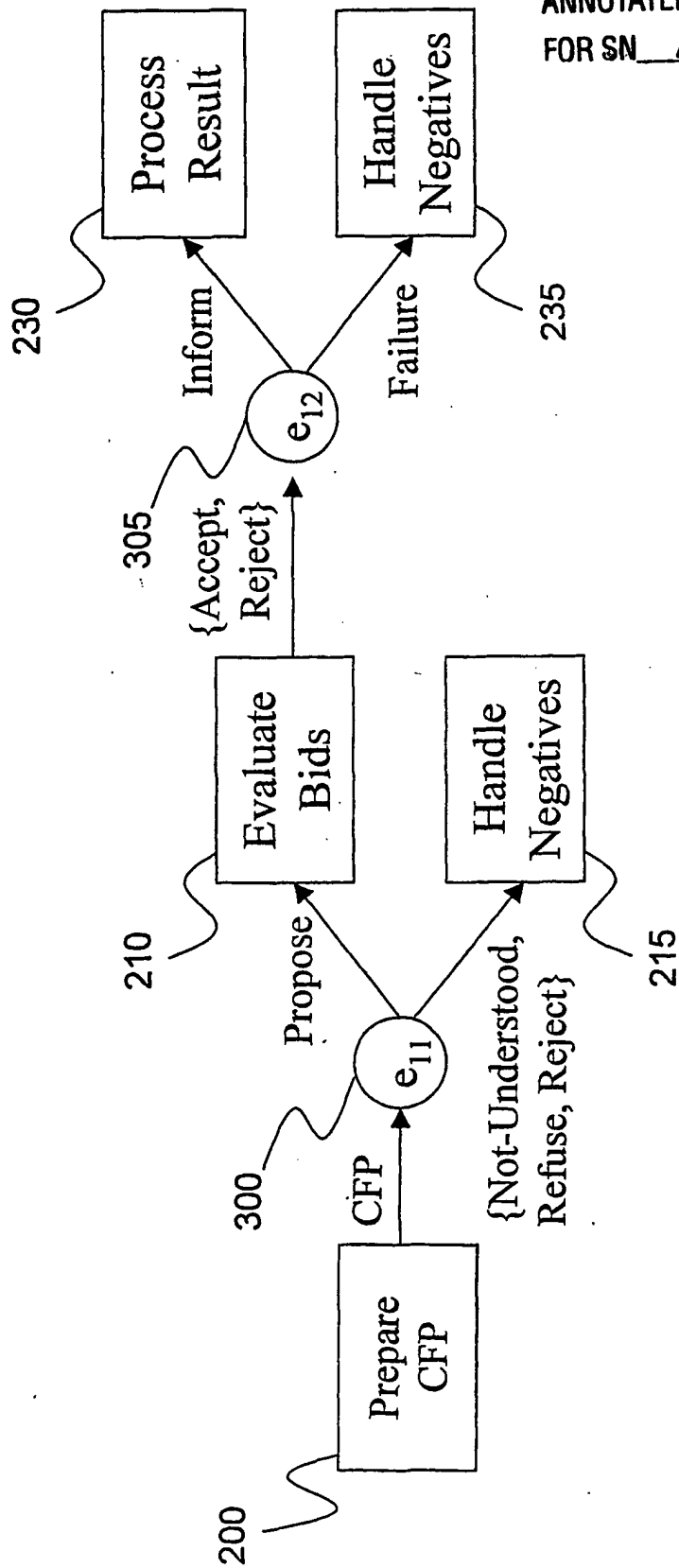


Figure 3

clean up size + symbolize

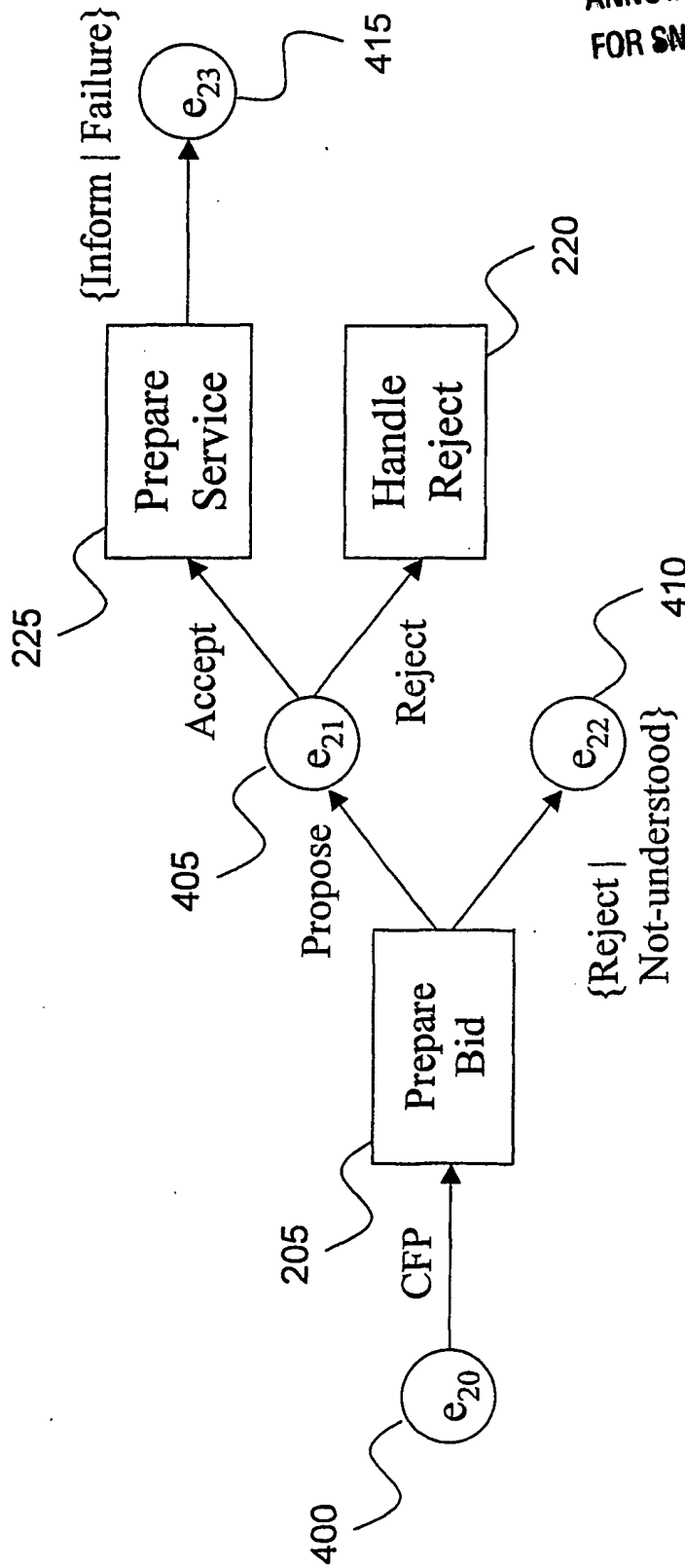


Figure 4

*clean up
& finalize
formatting*

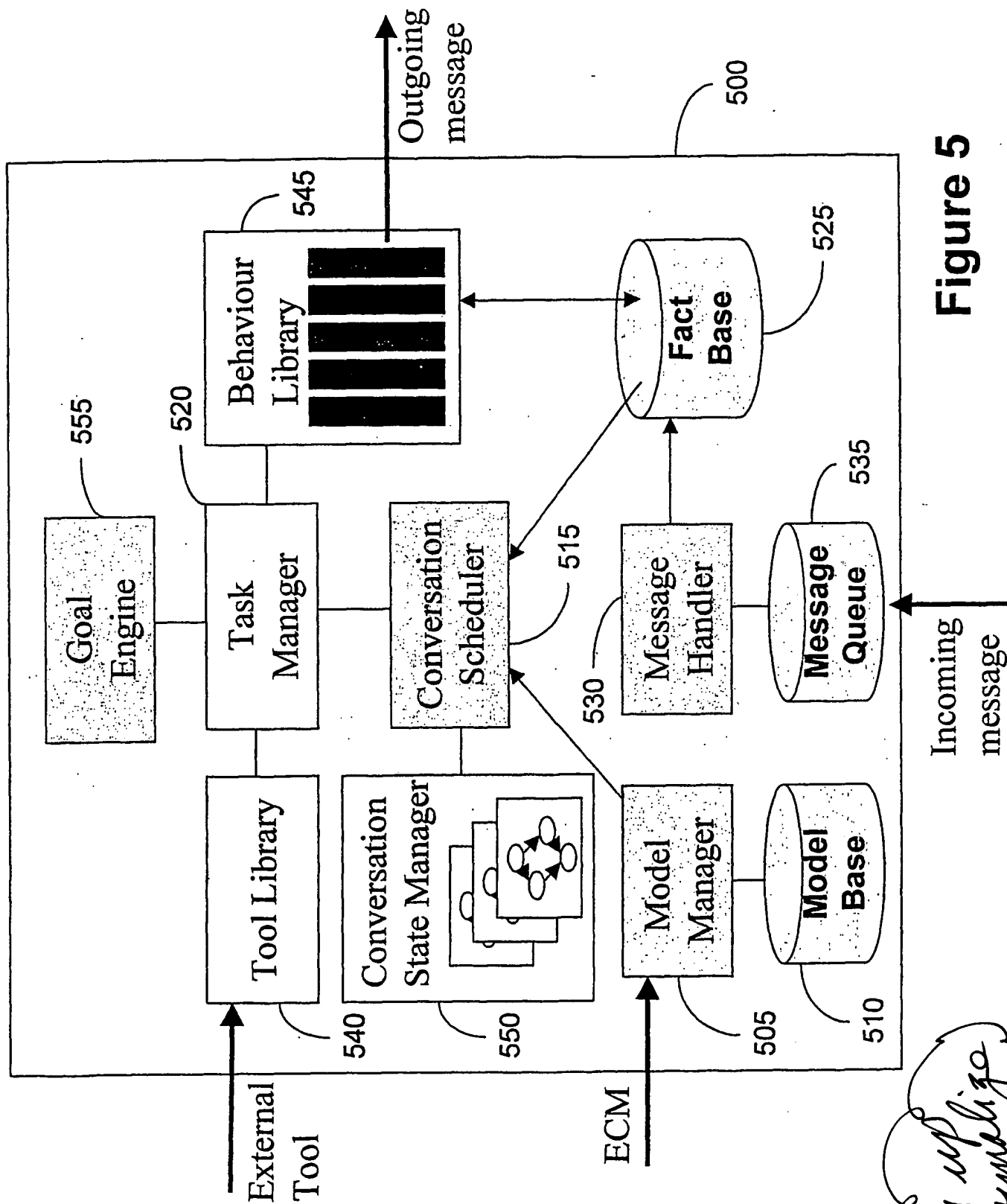


Figure 5

*clean up
+ formalize*

6/8

ANNOTATED MARKED UP DRAWINGS

FOR SN 10/594,424

START

Receive service request

Locate a mediator agent

Get service descriptions (SDs)

Any SD found?

Yes

Select an SD and
compare with
received service request

SD matches?

Yes

Is
a known CP
used?

No

Contact the mediator agent to
obtain a conversation model (CM)

Is the CM executable?

Yes

Execute the CM

Return service result

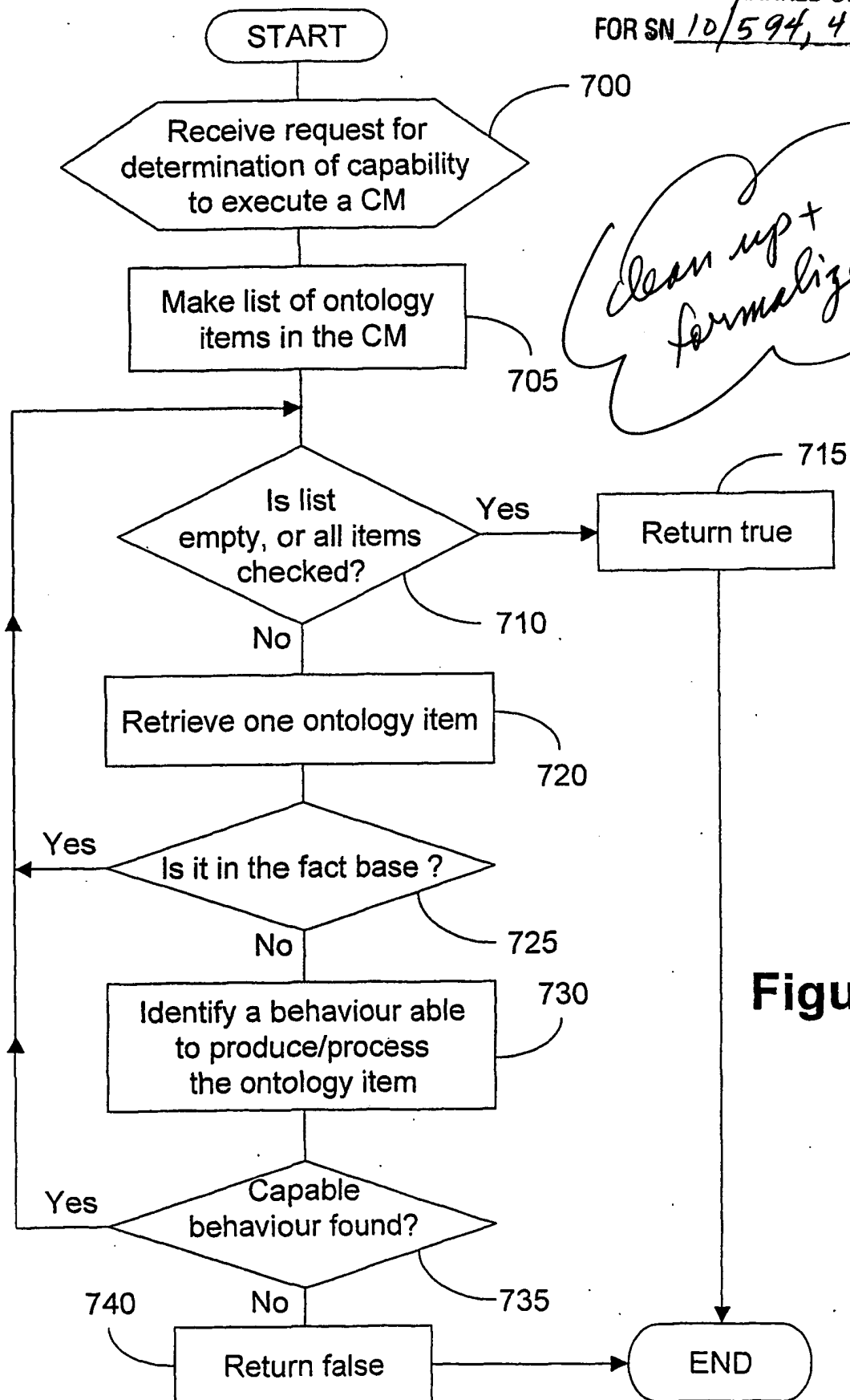
END

Try another
service
agent's SDNotify requester
of request failure

*clean up
+ formalize*

Figure 6

7/8.

ANNOTATED MARKED UP DRAWINGS
FOR SN 10/594,424**Figure 7**

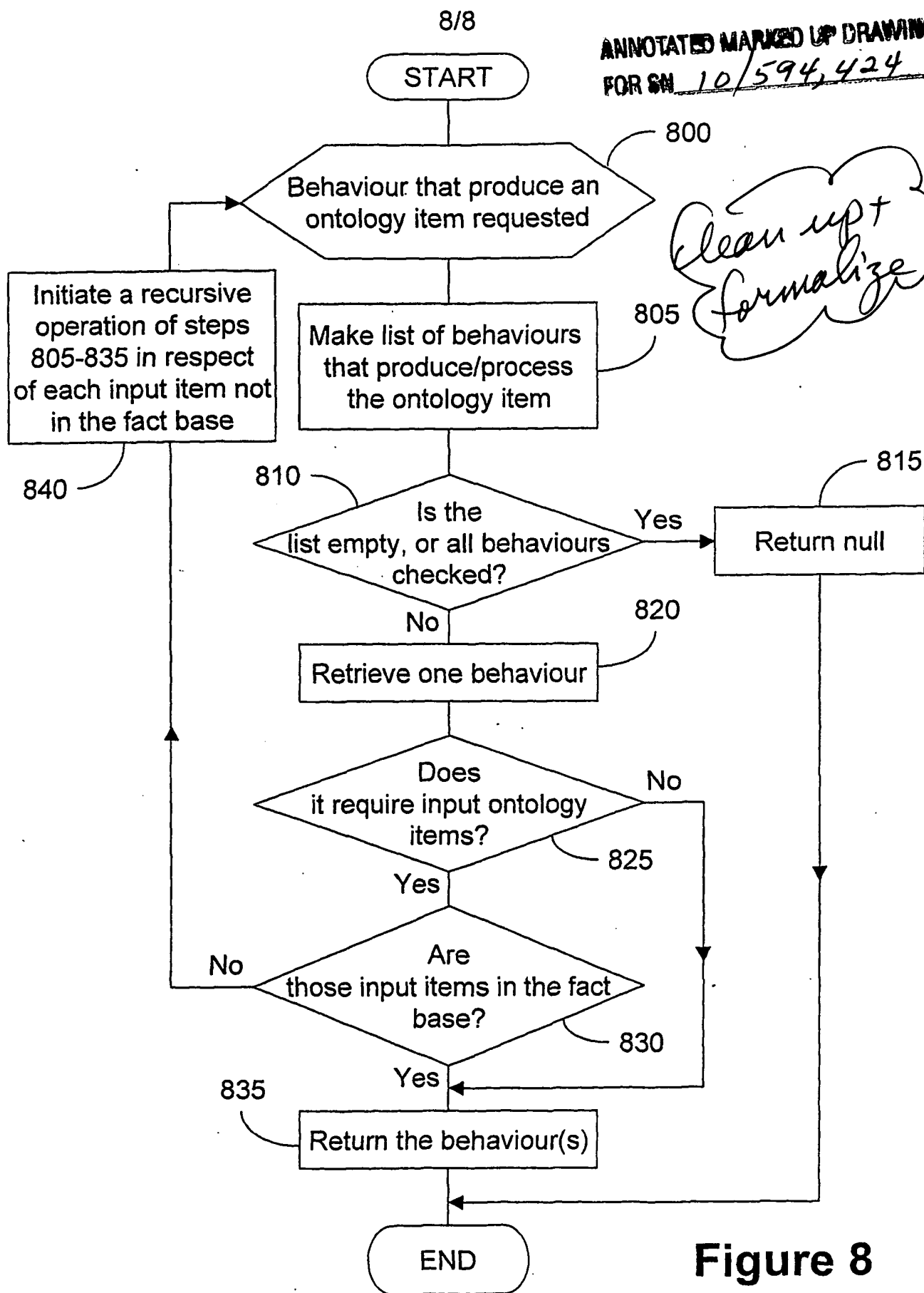


Figure 8